

# OpenAFS unter User Mode Linux



so geht's

Stephan Wiesand

DESY Zeuthen - DV -

10. Oktober 2003



# UML ?!

- (leicht) gepatchter **Linux Kernel**
- läuft als normaler **Prozess im Userland**
  - keine root-Rechte nötig
- **Block Devices** sind Dateien im Host-System
- **Netzwerkoptionen:**
  - rein virtuell (switch daemon)
  - Verbindung mit Host-Netzwerk via TUN/TAP
- <http://user-mode-linux.sourceforge.net>



# Wie geht's ?

- Zutaten:
  - Kernel Source
    - z.B. 2.4.22
    - an beliebiger Stelle auspacken
  - `uml-patch`
    - z.B. 2.4.22-5
  - Zubehör
    - `uml_utilities_20030903.tar`
    - `make; make install`
    - Ergebnis muss in `PATH` sein wenn Kernel gestartet wird



# Bau des Kernels

- nach dem Patchen:
  - `make xconfig ARCH=um`
    - Standardkonfiguration nach Patch ist ok
  - `make dep clean ARCH=um`
  - `make linux modules ARCH=um`
    - erzeugt Programm "linux"
    - Standardkonfiguration hat keine Module
- der Kernel kann jetzt gestartet werden
  - Panic, weil kein root-Dateisystem gefunden wird



# Dateisysteme

- einige sind **fertig** im Web **verfügbar**
  - meist etwas älter (RedHat <~ 7.2)
- einige Tools verfügbar (**mkrootfs**, UML-builder)
- oder man erzeugt **selbst** eines:
  - **Filesystem** auf sparse File erzeugen, loop mount
  - **RPMs** aus Minimalselektion installieren (chroot)
  - Hostname, hosts, fstab, Zeitzone, root user, Netzwerkkonfiguration, inittab, mknod, kleine **Hacks**
  - ca. 300 Zeilen Perl für SuSE 8.2, incl. FS/Scripts
  - ca. 2 Minuten



# Fstab

/dev/ubda	/	ext2	defaults	1	1
/dev/ubdb	swap	swap	defaults	0	0
/dev/ubdc	/afs_cache	ext2	defaults	1	2
/dev/ubdd	/vicepa	ext2	defaults	1	2
devpts	/dev/pts	devpts	mode=0620,gid=5	0	0
proc	/proc	proc	defaults	0	0
none	/host	hostfs	defaults	0	0

- **hostfs** gibt Zugriff auf / des Hosts
- ubda = (98, 0)
  - ubda1 = (98,1)
- ubdb = (98, 16)



# Start des Kernels

- für virtuelles Netzwerk: zunächst `uml_switch`
- `./linux \`
  - `ro \`
  - `devfs=nomount \`
  - `mem=64mb \`
  - `eth0=daemon,fe:fd:00:00:00:0b \`
  - `ubd0=root_fs \` # wird `/dev/ubda`
  - `ubd1=swap_file` # wird `/dev/ubdb`
- UML hängt noch `root=/dev/ubda` an

# Übersetzen von OpenAFS



- sysname: `i386_umlinux24`
  - `src/comerr/Makefile.in` `buglet`
    - `s/\*_linux\*/\*_linux\* | \*_umlinux\*/`
  - Übersetzen **im virtuellen Rechner**
    - geht wie auf einem realen Host, ist aber langsam
  - Übersetzen **im Hostsystem**
    - erfordert etwas Aufmerksamkeit um die korrekten Compileroptionen zu erhalten
- ```
•make GCC_KOPTS=" `(cd ../linux-2.4.22; make \
script 'SCRIPT=@echo $(CFLAGS)' ARCH=um) `"
```



# OpenAFS unter UML

- OpenAFS kann jetzt normal installiert werden
- ABER: es funktioniert nur der LWP-Fileserver
  - aus `src/viced`
  - wird immer mitgebaut
  - aber `dest/root.server` enthält im Normalfall den `pthreaded` Fileserver aus `src/tviced`